

JPWL: JPEG 2000 for Wireless Applications

Frederic Dufaux^{a,b}, and Didier Nicholson^c

^a Signal Processing Institute, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland
Frederic.Dufaux@epfl.ch

^b Emitall S.A., Rue du Théâtre 5, 1820 Montreux, Switzerland
Frederic.Dufaux@emitall.com

^c Thales Communications, 160 Blvd de Valmy, 92704 Colombes, France,
Didier.Nicholson@fr.thalesgroup.com

ABSTRACT

In this paper, we present the current status of the JPWL standardization work item. JPWL is an extension of the JPEG 2000 baseline specification in order to enable the efficient transmission of JPEG 2000 codestream over an error-prone network. In particular, JPWL supports a set of tools and methods for error protection and correction such as Forward Error Correcting (FEC) codes, Unequal Error Protection (UEP), and data partitioning and interleaving.

We then evaluate the performance of the JPWL Error Protection Block (EPB) tool. We consider two configurations of EPB: to protect the Main and Tile-part headers, or to protect the whole codestream using UEP. Experimental results show a significant quality improvement when using EPB compared to baseline JPEG 2000.

Keywords: JPEG 2000, JPWL, wireless imaging, error resilience, error correction

1. INTRODUCTION

Nowadays, multimedia wireless applications are becoming omnipresent. Indeed, recent reports have shown that sales of camera-equipped cell phones have exceeded those of standalone digital cameras. However, wireless networks are subject to frequent transmission errors. In this context, the development of efficient and robust wireless imaging solutions is important.

JPEG 2000 is the newest standard for still image coding [1][2]. On top of very high coding efficiency, JPEG 2000 also provides with a number of highly desirable features such as seamless progressive transmission by resolution or quality, lossy to lossless compression, random code stream access and processing, continuous-tone and bi-level compression, and region of interest. In [3], JPEG 2000 is reviewed and its performance analyzed and evaluated.

Given its qualities, JPEG 2000 is a leading contender for wireless multimedia applications. In addition, its high scalability enables a number of quality of service and revenue strategies. However, to succeed in this task, JPEG 2000 has to be robust to transmission errors.

The baseline JPEG 2000 defines error resilience tools to improve performances over noisy channels. A review is presented in [4], along with a performance comparison with MPEG-4 for still image coding. In [5], subjective results for Motion JPEG 2000 quality in mobile applications have been presented. In [6], a comparison of Motion JPEG 2000 and MPEG-4 in a wireless environment showed the good performance of Motion JPEG 2000.

However, the error resilience tools in baseline JPEG 2000 only detect whenever an error occurs, conceal the erroneous data, and resynchronize the decoder. In particular, they cannot correct transmission errors. Moreover, they do not address the presence of errors in the image header, even though it is the most important part of the codestream. For these reasons, the baseline JPEG 2000 error resilience tools are not sufficient in the context of wireless transmissions.

To overcome these limitations, JPEG has started a new work item referred to as Wireless JPEG 2000 or JPWL. The goal of JPWL is to extend the baseline JPEG 2000 specification to achieve the efficient transmission of JPEG 2000 imagery over an error-prone transmission environment. In particular, JPWL supports a set of tools for error protection and correction such as Forward Error Correcting (FEC) codes, Unequal Error Protection (UEP), data partitioning and interleaving, and robust arithmetic coding. Examples of tools envisioned to be supported by JPWL include the techniques presented in [7][8][9][10]. JPWL is not linked to a specific network or transport protocol, but provides a generic solution for the robust transmission of JPEG 2000 imagery over error-prone networks.

This paper is structured as follow. In Sec. 2, we first review the error resilience tools in the baseline JPEG 2000. We give an overview of JPWL in Sec. 3. We present experimental results evaluating the performance of the JPWL Error Protection Block (EPB) tool to protect JPEG 2000 codestream in Sec. 4. Finally, we draw some conclusions in Sec. 5.

2. ERROR RESILIENCE IN BASELINE JPEG 2000

The JPEG 2000 still image coding standard has recently been completed by the Joint Photographic Experts Group (JPEG) and approved as an International Standard in December 2000. The baseline version, also referred to as part 1 of the specifications, defines an efficient image coding scheme with compelling functionalities such as progressive coding up to lossless, seamless scalability, region of interest, random code stream access and processing, and continuous-tone and bi-level compression. In this section, we discuss the error resilience tools in the baseline JPEG 2000 specification. For a more detailed description of JPEG 2000, the reader is referred to [1][2].

Coding efficiency and robustness to transmission errors clearly define contradictory requirements. On the one hand, the goal of compression is to remove redundancies in the data in order to more efficiently represent it. On the other hand, error resilience aims at adding redundancies in the data in order to control the impact of transmission errors.

As most coding schemes, JPEG 2000 is using a Variable Length Codes (VLC). These codes are especially sensitive to transmission errors. Indeed, upon the occurrence of an error, the decoder usually loses synchronization and is unable to further decode. Consequently, not only the part of the bitstream where an error occurs is lost, but also all the data until the next position where the decoder is able to resynchronize. In addition, the effect of transmission errors will spread if decoding a part of the bitstream depends on the correct decoding of previous parts of the bitstream. Therefore, it is desirable to have coded units which can be independently decoded.

The error resilience tools in JPEG 2000 aim at detecting the occurrence of errors, concealing the erroneous data, and resynchronizing the decoder in order to limit the impact of transmission errors.

More specifically, the JPEG 2000 codestream is composed of independently coded units, also referred to as packets. Each packet corresponds to a quality layer, a resolution, a component and a precinct. Furthermore, resynchronization markers, also referred to as Start of Packet (SOP), can be optionally inserted in front of every packet, enabling the decoder to resynchronize in the presence of errors.

The wavelet coefficients are partitioned into code-blocks, and each code-block is independently coded using an MQ arithmetic coder. A number of options can be used to strengthen its robustness to errors. The arithmetic coder can be terminated and the contexts can be reset after each coding pass. Additionally, a segment marker can be encoded at the end of each coding pass. In this case, if the segment marker is not correctly decoded at the decoder side, an error is flagged in the preceding coding pass.

In the case of Motion JPEG 2000, as each frame is coded independently (intra-frame coding), transmission errors in one frame do not propagate to subsequent frames.

3. JPWL OVERVIEW

Given the importance of wireless imaging applications, JPEG recently kicked off a new activity referred to as Wireless JPEG 2000 or JPWL, also known formally as part 11 of the JPEG 2000 specifications. Its goal is to extend the baseline specification in order to allow for the efficient transmission of JPEG 2000 image data over an error-prone wireless transmission environment. More specifically, JPWL defines a set of tools and methods to protect the codestream against

transmission errors. It also defines means to describe the sensitivity of the codestream to transmission errors, and to describe the locations in the codestream of residual transmission errors. JPWL is notably addressing the protection of the image header, joint source-channel coding, unequal error protection, and data interleaving.

In this section, we review the current status of JPWL. As of April 2004, JPWL is at the Working Draft level [11]. Therefore, the description below is still subject to change.

3.1. Scope

The transmission of image and video content over wireless networks is becoming ubiquitous. Wireless networks are characterized by the frequent occurrence of transmission errors, which put strong constraints on the transmission of digital imagery. Given its high compression efficiency, JPEG 2000 is a very strong contender in wireless multimedia applications. Moreover, the highly scalable JPEG 2000 codestream enables a wide range of quality of service strategies for network operators.

However, JPEG 2000 has to be robust to transmission errors in order to be suitable for wireless imaging applications. While the baseline specification defines a number of tools for error resilience (see Sec. 2), these tools only detect the occurrence of errors, conceal the erroneous data and resynchronize the decoder. In particular, they do not correct transmission errors and do not address the occurrence of errors in the image header even though it is the most important part of the codestream. For these reasons, they are not sufficient in wireless imaging.

To overcome these limitations, JPWL extends the baseline specification and defines additional tools for error protection and correction. Examples of such tools are presented in [7][8][9][10]. JPWL is not only addressing the transmission of JPEG 2000 still images, but also the transmission of Motion JPEG 2000 video.

JPWL is not linked to a specific network or transport protocol, but provides a generic solution for the robust transmission of JPEG 2000 codestream over error-prone networks. While the main target of JPWL is wireless applications, the same tools can also be employed in other error-prone applications.

3.2. General JPWL system description

The main functionalities of the JPWL system are:

- to protect the codestream against transmission errors,
- to describe the degree of sensitivity of different parts of the codestream to transmission errors
- to describe the locations of residual errors in the codestream.

The JPWL system can either be applied to an input source image or to a JPEG 2000 codestream, as is illustrated in Figure 1 and Figure 2 respectively. In Figure 1, at the transmission side, a JPWL encoder consists of three modules running concurrently: a JPEG 2000 baseline encoder compressing the input image, a generator of the error sensitivity description, and a processor applying the error protection tool. The result is a JPWL codestream robust to transmission errors. At the receiving side, a JPWL decoder is also composed of three modules: a processor to correct errors, a generator of the residual errors description and a JPEG 2000 baseline decoder. Alternatively, in Figure 2, at the transmission side a JPWL transcoder processes a JPEG 2000 codestream, generating the error sensitivity description and applying error protection tools. At the receiving side, a JPWL transcoder corrects the transmission errors and generates the residual errors description, producing a JPEG 2000 codestream which can be sent to a JPEG 2000 baseline decoder, along with residual errors information.

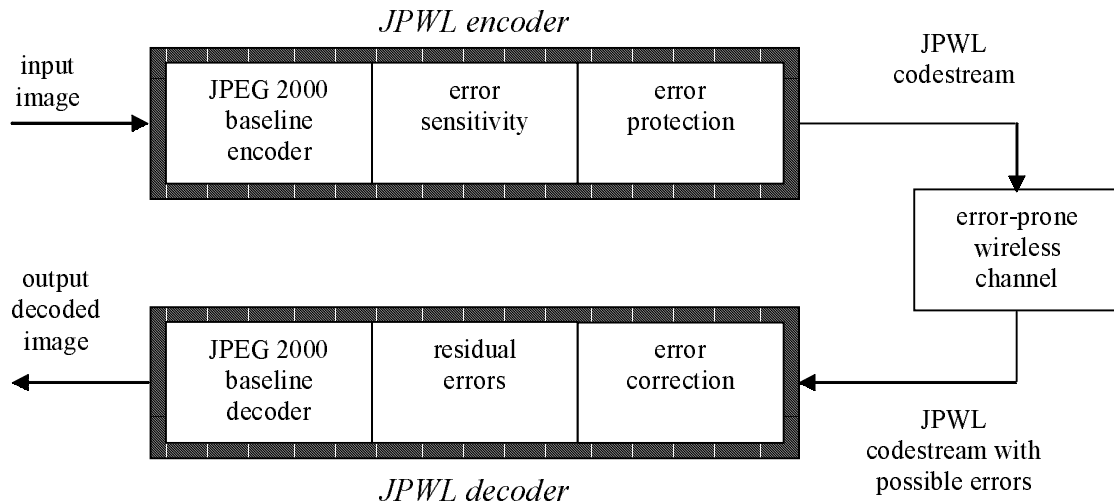


Figure 1 – JPWL system description: encoder and decoder.

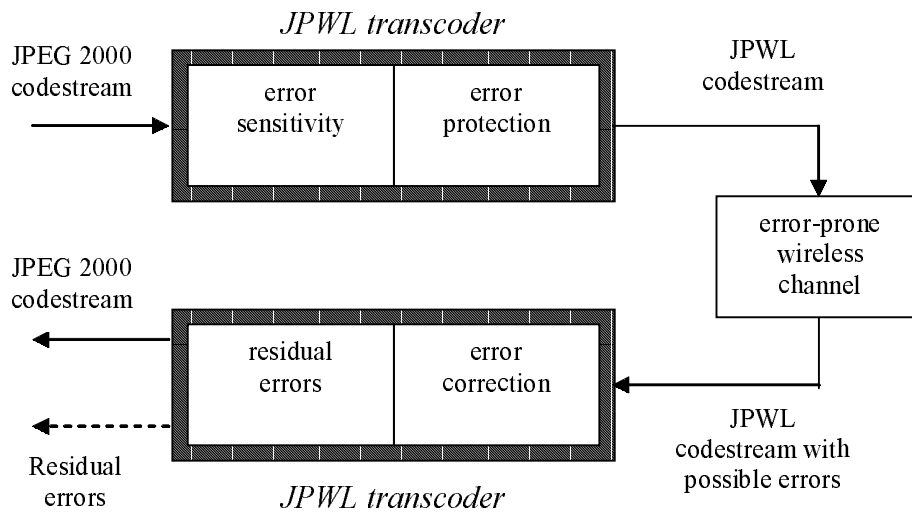


Figure 2 – JPWL system description: transcoder.

Other similar configurations are also possible. Whereas in Figure 1 and Figure 2 the generation of the error sensitivity description and the application of the error protection tool are concurrent, the two operations can be performed separately and independently. For instance, a first JPWL encoder/transcoder produces a JPWL codestream containing error sensitivity information. At a later time, a second JPWL transcoder uses this information to optimize the error protection tool, generating a JPWL codestream robust to transmission errors.

The error protection process modifies the codestream to make it more resilient to errors, e.g. by adding redundancy or by partitioning and interleaving the data. The error correction process detects the occurrence of errors and corrects them whenever possible. Techniques to protect the codestream include Forward Error Correcting (FEC) codes, data partitioning and interleaving, and Unequal Error Protection (UEP).

The specific tools for error protection have to be registered with the JPWL Registration Authority (RA). Upon registration, a tool is assigned a unique identification number. The corresponding JPWL syntax contains the unique

identification number along with private parameters. A JPWL application may have to query the JPWL RA in order to get a description of the tool and be able to correctly decode the image data. With this registration process, provision is made for future tools to be identified and registered.

The error sensitivity descriptor describes the degree of sensitivity of different parts of the codestream to transmission errors. This information is typically generated when the image is encoded using a JPEG 2000 baseline encoder (e.g. Figure 1), but it can also be directly derived from a JPEG 2000 codestream (e.g. Figure 2). This information can subsequently be used when protecting the image. More specifically, sensitive parts of the codestream can be more heavily protected than less sensitive parts (unequal error protection).

The residual errors descriptor specifies the locations of residual errors in the codestream. The residual errors are the errors which cannot be corrected by the error protection tool. This information is typically generated during the error correction process. This information can subsequently be used in the JPEG 2000 baseline decoder to prevent decoding corrupted parts of the stream.

3.3. Syntax

In this section, we present the current JPWL syntax. Four new marker segments have been defined by JPWL, Error Protection Capability (EPC), Error Protection Block (EPB), Error Sensitivity Descriptor (ESD) and Residual Error Descriptor (RED). They are described in more details hereafter.

3.3.1. Error Protection Capability (EPC)

The EPC marker segment indicates which JPWL normative and informative tools are used in the codestream. More specifically, EPC signals whether the three other normative marker segments defined by JPWL, namely the error sensitivity descriptor (ESD), the residual error descriptor (RED) and the error protection block (EPB) are present in the codestream.

Furthermore, EPC signals the use of informative tools which have been previously registered with the JPWL RA. Upon registration, each tool is assigned an ID, which uniquely identifies it. These informative tools allow for error resilience and/or error correction, and include techniques such as error resilient entropy coding, Forward Error Correcting (FEC) codes, Unequal Error Protection (UEP), data partitioning and interleaving. EPC may also contain parameters relative to these informative tools. Therefore, this syntax allow for a flexible use of existing tools and the rollout of new ones in the future.

When encountering a JPWL codestream, the decoder can identify the tool(s) which have been used to protect this codestream by parsing the EPC marker segment and by querying the JPWL RA. The decoder can then take the appropriate steps to decode the codestream, e.g. acquire or download the appropriate tool.

The EPC marker segment syntax is illustrated in Figure 3.

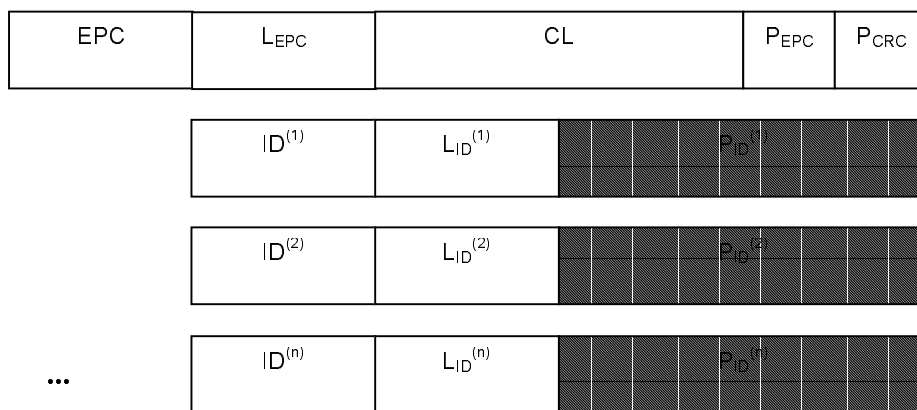


Figure 3 – Syntax for the EPC marker segment.

EPC is a unique marker which identifies the EPC marker segment, its value is 0xFF97. L_{EPC} is the length of the marker segment in bytes. CL indicates the total codestream length in bytes. P_{EPC} signals the presence of the ESD, RED and EPB marker segments, and the use of informative techniques in the codestream. P_{CRC} are parity check bits to verify that the EPC marker segment is not corrupted. $ID^{(i)}$ is the identification number issued by the JPWL RA for tool i, $P_{ID}^{(i)}$ contains parameters for tool i, and $L_{ID}^{(i)}$ is the length of $L_{ID}^{(i)} + P_{ID}^{(i)}$ in bytes.

3.3.2. Error Protection Block (EPB)

The primary function of EPB is to protect the Main and Tile-part header. However, it can also be used to protect the remaining of the bitstream. The EPB marker segment contains information about the error protection parameters and redundancy data used to protect the codestream against errors [8].

The syntax of the EPB marker segment is shown in Figure 4.

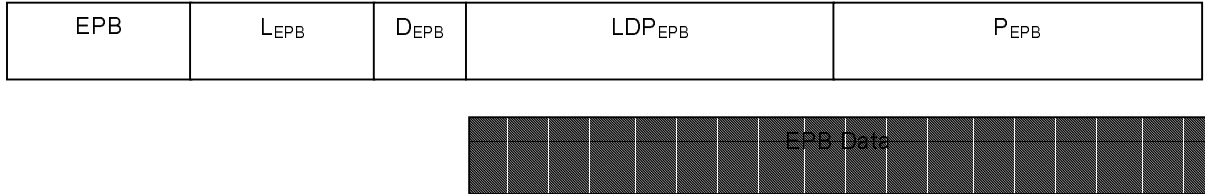


Figure 4 – Syntax for the EPB marker segment.

EPB is a unique marker which identifies the EPB marker segment, its value is 0xFF96. L_{EPB} is the length of the marker segment in bytes. D_{EPB} specifies the usage of EPB. LDP_{EPB} is the length of the data to be protected by the redundant information carried within the current EPB marker segment. P_{EPB} allows changing the error correcting code used in the remaining data. Finally, EPB Data contains the data to perform error correction, typically redundancy bits.

There can be one or more EPB marker segments in the Main header and/or Tile-part headers. In order to enable resynchronization in the presence of errors, the first EPB marker segment in a Main header is required to be placed immediately after the SIZ marker segment, and the first EPB marker segment in a Tile-part header is required to be placed immediately after the SOT marker.

The first EPB marker segment of the Main header is protected using a Reed Solomon RS(160,64) code. The first EPB marker segment of a Tile-part header is protected using RS(80,25) code. Finally, by default a predefined RS(40,13) code is used for the other EPB marker segment. The latter can be changed using the P_{EPB} parameter.

3.3.3. Error Sensitivity Descriptor (ESD)

The ESD marker segment contains information about the sensitivity of codestream to errors. This information can be exploited when applying an Unequal Error Protection (UEP) technique. Straightforwardly, more powerful codes are used to protect the most sensitive portion of the codestream. This information can also be used for selective retransmissions as proposed in [10]. More specifically, a larger number of retransmissions are attempted for the most critical parts of the codestream. Finally, the information carried in ESD could also be used for other non-JPWL applications such as efficient rate transcoding or smart prefetching.

Note that the usage of the error sensitivity information is not specified by JPWL. Moreover, this information is not required in order to decode a codestream.

The syntax of the ESD marker segment is given in Figure 5.

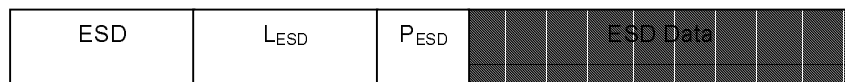


Figure 5 – Syntax for the ESD marker segment.

ESD is a unique marker which identifies the ESD marker segment, its value is 0xFF98. L_{ESD} is the length of the marker segment in bytes. P_{ESD} specifies the usage of ESD. For instance, ESD can use a byte-range mode, a packet mode or a packet-range mode. Also, error sensitivity data can be expressed in different ways, such as relative values, Mean Square Errors (MSE) or Peak-Signal-Noise-Ratio (PSNR). Finally, ESD Data contains the record of error sensitivity data itself.

3.3.4. Residual Error Descriptor (RED)

The RED marker segment signals the presence of residual errors in the codestream. Indeed, a JPWL decoder may fail to correct all the errors in a codestream. RED allows signaling the location of such residual errors. This information can then be exploited by a JPEG 2000 decoder in order to better cope with errors. For instance, the decoder could request retransmission, conceal the errors or discard the corrupted information.

Note that the usage of the residual error information is not specified by JPWL. Moreover, this information is not required in order to decode a codestream.

Figure 6 illustrates the syntax of the RED marker segment.

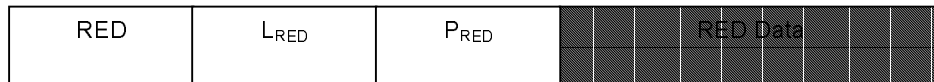


Figure 6 – Syntax for the RED marker segment.

RED is a unique marker which identifies the RED marker segment, its value is 0xFF99. L_{RED} is the length of the marker segment in bytes. P_{RED} specifies the usage of RED. In particular, RED can use a byte-range mode, a packet mode or a packet-range mode. Finally, RED Data contains the record of residual error data itself.

3.4. Normative and informative parts

JPWL defines a normative codestream syntax which specifies the information required for interpreting the JPWL tools which have been applied to the codestream. JPWL also defines a normative process for registering error protection tools with the JPWL RA.

The specific tools to protect an image are out of the scope of JPWL. Instead, JPWL gives informative examples of typical use cases, along with guidelines on how to best implement error protection and correction.

A JPWL compliant application implements one or more JPWL tools and is therefore able to consume JPWL codestreams.

4. SIMULATION RESULTS

In this section, we present simulation results for the transmission of Motion JPEG 2000 video over an error-prone channel. In particular, we evaluate the efficiency of the error resilience tools in the baseline JPEG 2000 coding scheme, and the performance of the JPWL Error Protection Block (EPB) tool described in Sec. 3.3.2. More precisely, we consider two configurations of the EPB tool, in the first case it is used to protect the Main and Tile-part headers, whereas in the second case it is applied to protect the whole codestream using Unequal Error Protection (UEP).

The system depicted in Figure 7 is used to simulate video transmission over a WCDMA wireless channel. The video source is first encoded with JPEG 2000, using the Kakadu software [12]. The JPWL EPB tool is optionally applied in order to protect the codestream. We then use a WCDMA error pattern in order to simulate the random occurrence of transmission errors. As the injection of transmission errors is a random process, 50 trials are run for each simulation case and final results are the average over all the trials. For each trial, a different random circular shift is applied to the same error pattern file. The output video is obtained by applying the inverse operations, namely the optional JPWL EPB decoding followed by JPEG 2000 decoding.

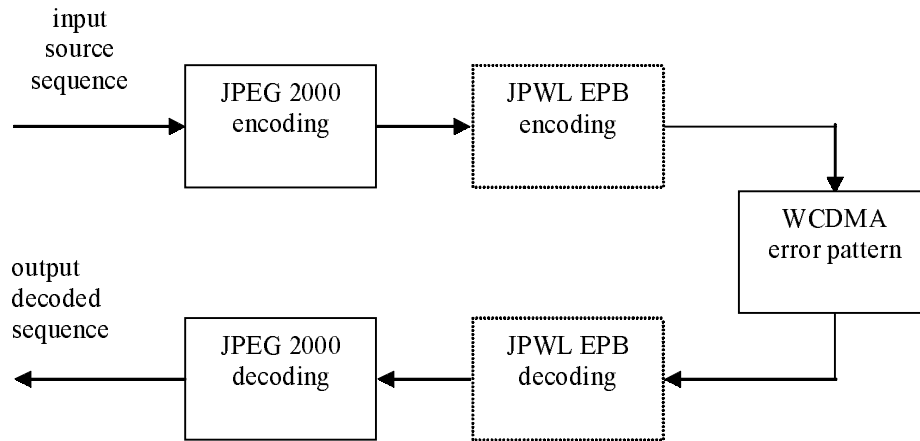


Figure 7 – Simulation environment for the transmission of video over WCDMA.

The test sequences City, Crew, Foreman, Harbor, Mobile and Soccer have been used in our simulation. These sequences exhibit very different characteristics. The first frame of each sequence is depicted in Figure 8. The sequences are in CIF format with a frame rate of 15 fps. The sequences are encoded at 384 kb/s. The transmission errors have a Bit Error Rate (BER) of $1e-3$.



Figure 8 – Test sequences: City, Crew, Foreman, Harbor, Mobile and Soccer.

We compare the four following cases:

1. Baseline JPEG 2000 encoding without error resilience tools; we use 3 quality layers, a code-block size of 64x64 and 4 levels of wavelet decomposition
2. Baseline JPEG 2000 encoding with the following baseline error resilience tools:
 - RESTART: The MQ coder is restarted at the beginning of each coding pass
 - ERTerm: The encoder enforces a predictable termination policy for the MQ coder.
 - SEGMark: A special symbol is encoded at the end of each bit-plane
 - SOP: Start of Packet markers are inserted in front of every packet
3. Same as 2, with the addition of JPWL EPB in order to protect the Main Header and Tile-part Header with the following codes:
 - RS(160,64) code is used to protect the first EPB marker segment of the Main header
 - RS(80,25) code is used to protect the first EPB marker segment of a Tile-part header
 - RS(40,13) code is used to protect the remaining EPB marker segments in Main and Tile-part header
4. Same as 2, with the addition of JPWL EPB in order to protect the whole codestream using UEP as follow:
 - RS(160,64) code is used to protect the first EPB marker segment of the Main header
 - RS(80,25) code is used to protect the first EPB marker segment of a Tile-part header
 - RS(40,13) code is used to protect the remaining EPB marker segments in Main and Tile-part header
 - RS(30,20) code is used to protect the first layer
 - RS(26,20) code is used to protect the second layer
 - The third layer is not protected

Table 1 summarizes the results expressed in terms of Peak-Signal-Noise-Ratio (PSNR).

sequence	JPEG 2000 baseline	JPEG 2000 baseline + error resilience	JPEG 2000 baseline + error resilience + JPWL EPB-HEADER	JPEG 2000 baseline + error resilience + JPWL EPB-UEP
City	23.61	24.98	26.36	26.65
Crew	26.12	28.09	29.66	30.43
Foreman	23.56	25.73	27.22	28.07
Harbor	19.74	21.19	22.31	22.60
Mobile	16.68	17.71	18.67	18.63
Soccer	25.06	27.35	28.96	29.80
average	22.46	24.18	25.53	26.03

Table 1 – PSNR results.

As can be observed, the error resilience tools result in a gain of 1.72 dB on average when compared to baseline without error resilience. The additional use of EPB leads to even more significant quality improvement. When EPB is applied on the Main and Tile-part headers, the gain achieved is on average 3.07 dB compared to baseline without error resilience, and 1.35 dB against baseline with error resilience. When EPB is applied on the whole codestream using UEP, the

performance is on average 3.57 dB higher than baseline without error resilience, and 1.85 dB higher than base line with error resilience.

Furthermore, when EPB is not used, errors often occur in the Main or Tile-part headers leading to frequent decoder crashes. When EPB is used, those errors can usually be successfully corrected, hence greatly reducing the number of decoder crashes.

5. CONCLUSIONS

In this paper, we have first reviewed the on-going JPWL standardization effort. The goal of JPWL is to define an extension of the JPEG 2000 baseline specification in order to enable the efficient transmission of JPEG 2000 codestream over an error-prone network. JPWL supports a set of tools for error protection and correction such as Forward Error Correcting (FEC) codes, Unequal Error Protection (UEP), data partitioning and interleaving, and robust arithmetic coding.

We have then evaluated the performance of the JPWL Error Protection Block (EPB) tool. We considered two configurations of EPB: to protect the Main and Tile-part headers, or to protect the whole codestream using UEP. Experimental results showed a significant quality improvement when using EPB. Gains up to 3.57 dB have been obtained when compared to JPEG 2000 baseline without error resilience, and up to 1.85 dB against baseline with error resilience tools.

ACKNOWLEDGEMENT

This work was partially supported by the European Project WCAM <http://www.ist-wcam.org> (IST Contract 507204) and the European Network of Excellence VISNET <http://www.visnet-noe.org> (IST Contract 506946), both funded under the European Commission IST 6th Framework Program.

The authors would like to thank all the individuals who participated to the JPWL ad-hoc group activities for their contributions to the development of JPWL.

REFERENCES

- [1] A. Skodras, C. Christopoulos and T. Ebrahimi, "The JPEG 2000 still image compression standard", IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 36-58, Sept. 2001.
- [2] D. Taubman and M. Marcellin, "JPEG 2000: Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.
- [3] D. Santa-Cruz, R. Grosbois and T. Ebrahimi, "JPEG 2000 performance evaluation and assessment", Signal Processing: Image Communication, vol. 17, no. 1, pp. 113-130, January 2002.
- [4] I. Moccagatta, S. Soudagar, J. Liang and H. Chen, "Error-Resilient Coding in JPEG-2000 and MPEG-4", IEEE Journal on Selected Areas in Communications, vol. 18, no. 6, pp. 899-914, June 2000.
- [5] S. Winkler and F. Dufaux, "Video Quality Evaluation for Mobile Applications", in SPIE Proc. Visual Communication and Image Processing, Lugano, Switzerland, July 2003.
- [6] F. Dufaux and T. Ebrahimi, "Error-Resilient Video Coding Performance Analysis of Motion JPEG 2000 and MPEG-4", SPIE Proc. Visual Communication and Image Processing, San Jose, CA, January 2004.
- [7] A. Natsu and D. Taubman, "Unequal protection of JPEG 2000 codestreams in wireless channels", in IEEE Proc. Globecom'02, Taipei, Taiwan, Nov. 2002.
- [8] D. Nicholson, C. Lamy-Begot, X. Naturel and C. Poulliat, "JPEG 2000 backward compatible error protection with Reed-Solomon codes", IEEE Trans. on Consumer Electronics, vol. 49, no. 4, pp. 855-860, Nov. 2003.
- [9] F. Frescura, M. Giorni, C. Feci and S. Cacopardi, "JPEG 2000 and MJPEG 2000 transmission in 802.11 wireless local area networks", IEEE Trans. on Consumer Electronics, vol. 49, no. 4, pp. 861-871, Nov. 2003.
- [10] M. Grangotto, E. Magli and G. Olmo, "Error sensitivity data structures and retransmission strategies for robust JPEG 2000 wireless imaging", IEEE Trans. on Consumer Electronics, vol. 49, no. 4, pp. 872-882, Nov. 2003.
- [11] "JPEG 2000 image coding system – Part 11: Wireless JPEG 2000 – Working Draft version 3.1", ISO/IEC JTC1/SC29/WG1 WG1N3294, April 2004.
- [12] <http://www.kakadusoftware.com> (v4.0).